SIAM J. Imaging Sciences
Vol. 11, No. 3, pp. 2001–2020

Plug-and-Play Unplugged: Optimization-Free Reconstruction Using Consensus Equilibrium[*]

Gregery T. Buzzard[†], Stanley H. Chan[‡], Suhas Sreehari[§], and Charles A. Bouman[¶]

Abstract. Regularized inversion methods for image reconstruction are used widely due to their tractability and their ability to combine complex physical sensor models with useful regularity criteria. Such methods motivated the recently developed Plug-and-Play prior method, which provides a framework to use advanced denoising algorithms as regularizers in inversion. However, the need to formulate regularized inversion as the solution to an optimization problem limits the expressiveness of possible regularity conditions and physical sensor models. In this paper, we introduce the idea of consensus equilibrium (CE), which generalizes regularized inversion to include a much wider variety of both forward (or data fidelity) components and prior (or regularity) components without the need for either to be expressed using a cost function. CE is based on the solution of a set of equilibrium equations that balance data fit and regularity. In this framework, the problem of MAP estimation in regularized inversion is replaced by the problem of solving these equilibrium equations, which can be approached in multiple ways. The key contribution of CE is to provide a novel framework for fusing multiple heterogeneous models of physical sensors or models learned from data. We describe the derivation of the CE equations and prove that the solution of the CE equations generalizes the standard MAP estimate under appropriate circumstances. We also discuss algorithms for solving the CE equations, including a version of the Douglas–Rachford/alternating direction method of multipliers algorithm with a novel form of preconditioning and Newton's method, both standard form and a Jacobian-free form using Krylov subspaces. We give several examples to illustrate the idea of CE and the convergence properties of these algorithms and demonstrate this method on some toy problems and on a denoising example in which we use an array of convolutional neural network denoisers, none of which is tuned to match the noise level in a noisy image but which in consensus can achieve a better result than any of them individually.

Key words. Plug-and-Play, regularized inversion, ADMM, tomography, denoising, MAP estimate, multiagent consensus equilibrium, consensus optimization

AMS subject classifications. 94A08, 68U10

[*] Received by the editors March 24, 2017; accepted for publication (in revised form) June 20, 2018; published electronically September 4, 2018.

http://www.siam.org/journals/siims/11-3/M112245.html

Funding: The work of the first author was partially supported by NSF grant DMS-1318894. The work of the second author was partially supported by NSF grant CCF-1718007. The work of the first, second, and fourth authors was partially supported by NSF grant CCF-1763896. The work of the third and fourth authors was partially supported by an AFOSR/MURI grant FA9550-12-1-0458, by UES Inc. under the Broad Spectrum Engineered Materials contract, and by the Electronic Imaging component of the ICMD program of the Materials and Manufacturing Directorate of the Air Force Research Laboratory, Andrew Rosenberger, program manager.

[†] Department of Mathematics, Purdue University, West Lafayette, IN 47907 (buzzard@purdue.edu).

[‡] School of Electrical and Computer Engineering and Department of Statistics, Purdue University, West Lafayette, IN 47907 (stanchan@purdue.edu).

[§] School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907 (ssreehar@purdue.edu).

[¶] School of Electrical and Computer Engineering and Weldon School of Biomedical Engineering, Purdue University, West Lafayette, IN 47907 (bouman@purdue.edu).

DOI. 10.1137/17M1122451

1. Introduction.

Over the past 30 years, statistical inversion has evolved from an interesting theoretical idea to a proven practical approach. Most statistical inversion methods are based on the maximum a posteriori (MAP) estimate, or more generally regularized inversion, using a Bayesian framework, since this approach balances computational complexity with achievable image quality. In its simplest form, regularized inversion is based on the solution of the optimization problem

$$x^* = \underset{x}{\operatorname{argmin}} \{f(x) + h(x)\}, \tag{1}$$

where f is the data fidelity function and h is the regularizing function. In the special case of MAP estimation, f represents the forward model and h represents the prior model, given by

$$f(x) = -\log p_{\text{forward}}(y|x), \qquad h(x) = -\log p_{\text{prior}}(x),$$

where y is the data and x is the unknown to be recovered. The solution of equation (1) balances the goals of fitting the data while also regularizing this fit according to the prior.

In more general settings, for example with multiple data terms from multimodal data collection, a cost function can be decomposed as a sum of auxiliary (usually convex) functions:

$$\text{minimize } f(x) = \sum_{i=1}^{N} f_i(x),$$

with variable $x \in \mathbb{R}^n$ and $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$. In consensus optimization, the minimization of the original cost function is reformulated as the minimization of the sum of the auxiliary functions, each a function of a separate variable, with the constraint that the separate variables must share a common value:

$$\text{minimize } \sum_{i=1}^{N} f_i(x_i) \text{ subject to } x_i = x, \ i = 1, \dots, N, \tag{2}$$

with variables $x \in \mathbb{R}^n$, $x_i \in \mathbb{R}^n$, $i = 1, \dots, N$. This reformulation allows for the application of the alternating direction method of multipliers (ADMM) or other efficient minimization methods and applies to the original problem in (1) as well as many other problems. An account of this approach with many variations and examples can be found in [4].

While regularized inversion and optimization problems more generally benefit from extensive theoretical results and powerful algorithms, they are also expressively limited. For example, many of the best denoising algorithms cannot be put into the form of a simple optimization; see [6, 10]. Likewise, the behavior of denoising neural networks cannot generally be captured via optimization. These successful approaches to inverse problems lie outside the realm of optimization problems and give rise to the motivating question for this paper.

Question. How can we generalize the consensus optimization framework in (2) to encompass models and operators that are not associated with an optimization problem, and how can we find solutions efficiently?

There is a vast and quickly growing literature on methods and results for convex and consensus optimization. Seminal work in this area includes the work of Lions and Mercier [19], as well as the PhD thesis of Eckstein [11] and the work of Eckstein and Bertsekas [12]. We do not provide a complete survey of this literature since our focus is on a framework beyond optimization, but some starting points for this area are [2, 4, 5].

As for approaches to fuse a data fidelity model with a denoiser that is not based on an optimization problem, the first attempt, to our knowledge, is [29]. The goal of this approach, called the Plug-and-Play prior method, is to replace the prior model in the Bayesian formulation with a denoising operator. This is done by taking the ADMM algorithm, which is often used to find solutions for consensus optimization problems, and replacing one of the optimization steps (proximal maps) of this algorithm with the output of a denoiser. Recently, a number of authors have built on the Plug-and-Play method as a way to construct implicit prior models through the use of denoising operators; see [25, 28, 27, 30]. In [28], conditions are given on the denoising operator that will ensure it is a proximal mapping, so that the MAP estimate exists and the ADMM algorithm converges. However, these conditions impose relatively strong symmetry conditions on the denoising operator that may not occur in practice. For applications where fixed point convergence is sufficient, it is possible to relax the conditions on the denoising operator by iteratively controlling the step size in the proximal map for the forward model and the noise level for the denoiser [7].

The paper [24] provides a different approach to building on the idea of Plug-and-Play. That paper uses the classical forward model plus the prior model in the framework of optimization but constructs a prior term directly from the denoising engine; this is called regularization by denoising (RED). For a denoiser $x \mapsto H(x)$, the prior term is given by $\lambda x^T(x - H(x))$. This approach is formulated as an optimization problem associated with any denoiser, but in the case that the denoiser itself is obtained from a prior, the RED prior is different from the denoiser prior; see [22]. Other approaches that build on Plug-and-Play include [21], which uses primal-dual splitting in place of an ADMM approach, and [17], which uses fast iterative shrinkage-thresholding algorithm in a Plug-and-Play framework to address a nonlinear inverse scattering problem.

In this paper, we introduce consensus equilibrium (CE) as an optimization-free generalization of regularized inversion and consensus optimization that can be used to fuse multiple sources of information implemented as maps such as denoisers, deblurring maps, data fidelity maps, proximal maps, etc. We show that CE generalizes consensus optimization problems in the sense that if the defining maps are all proximal maps associated with convex functions, then any CE solution is also a solution to the corresponding consensus optimization problem. However, CE can still exist in the more general case when the defining maps are not proximal maps; in this case, there is no underlying optimization. In the case of a single data fidelity term and a single denoiser, the solution has the interpretation of achieving the best denoised inverse of the data. That is, the proximal map associated with the forward model pulls the current point towards a more accurate fit to data, while the denoising operator pulls the current point towards a "less noisy" image. We illustrate this in a toy example in two dimensions: the CE is given by a balance between two competing forces.

In addition to introducing the CE equations, we discuss ways to solve them and give several examples. We describe a version of the Douglas–Rachford (DR)/ADMM algorithm with a novel form of anisotropic preconditioning. We also apply Newton's method, both in standard form and in a Jacobian-free form using Krylov subspaces.

In the experimental results section, we give several examples to illustrate the idea of CE and the convergence properties of these algorithms. We first demonstrate the proposed algorithms on some toy problems in order to illustrate properties of the method. We next use the CE framework to solve an image denoising problem using an array of convolutional neural network (CNN) denoisers, none of which is tuned to match the noise level in a noisy image. Our results demonstrate that that the CE result is better than any of the individually applied CNN denoisers.

2. Consensus equilibrium: Optimization and beyond.

In this section we formulate the CE equations, show that they encompass a form of consensus optimization in the case of proximal maps, and describe the ways that CE goes beyond the optimization framework.

2.1. Consensus equilibrium for proximal maps.

We begin with a slight generalization of (2):

$$\text{minimize} \sum_{i=1}^{N} \mu_i f_i(x_i) \text{ subject to } x_i = x,\ i = 1, \dots, N, \tag{3}$$

with variables $x \in \mathbb{R}^n$, $x_i \in \mathbb{R}^n$, $i = 1, \dots, N$, and weights $\mu_i > 0$, $i = 1, \dots, N$, that sum to 1 (an arbitrary normalization, but one that supports the idea of weighted average that we use later). From the point of view of optimization, each weight μ_i could be absorbed into f_i. However, in CE we move beyond this optimization framework to the case in which the f_i may be defined only implicitly or the case in which there is no optimization, but only mappings that play a role similar to the proximal maps that arise in the ADMM approach to solving (3). The formulation in (3) serves as motivation and the foundation on which we build.

To extend the optimization framework of (3) to CE, we start with N vector-valued maps, $F_i : \mathbb{R}^n \to \mathbb{R}^n$, $i = 1, \dots, N$. The CE for these maps is defined as any solution $(x^*, \mathbf{u}^*) \in \mathbb{R}^n \times \mathbb{R}^{nN}$ that solves the equations

$$F_i(x^* + u_i^*) = x^*,\ i = 1, \dots, N, \tag{4}$$

$$\bar{\mathbf{u}}_\mu^* = 0. \tag{5}$$

Here $\mathbf{u}$ is a vector in $\mathbb{R}^{nN}$ obtained by stacking the vectors $u_1, \dots, u_N$, and $\bar{\mathbf{u}}_\mu$ is the weighted average $\sum_{i=1}^N \mu_i u_i$.

In order to relate CE to consensus optimization, first consider the special case in which each $f_i : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ in (3) is a proper, closed, convex function and each F_i is a corresponding proximal map, i.e., a map of the form

$$F_i(x) = \underset{v}{\operatorname{argmin}} \left\{ \frac{\|v - x\|^2}{2\sigma^2} + f_i(v) \right\}. \tag{6}$$

Methods such as ADMM, DR, and other variants of the proximal point algorithm apply these maps in sequence or in parallel with well-chosen arguments, together with some map to promote $x_i = z$ for all i, in order to solve (3); see, e.g., [3, 4, 9, 11, 12, 19, 26]. In the setting of

Bayesian regularized inversion, each f_i represents a data fidelity term or regularizing function. We allow for the possibility that f_i enforces some hard constraints by taking on the value $+\infty$.

Our first theorem states that when the maps F_i are all proximal maps as described above, the solutions to the CE problem are exactly the solutions to the consensus optimization problem of equation (3). In this sense, CE encompasses the optimization framework of (3).

Theorem 1. *For $i = 1, \dots, N$, let f_i be a proper, lower-semicontinuous, convex function on $\mathbb{R}^n$, and let $\mu_i > 0$ with $\sum_{i=1}^N \mu_i = 1$. Define $f = \sum_{i=1}^N \mu_i f_i$, and assume f is finite on some open set in $\mathbb{R}^n$. Let F_i be the proximal maps as in (6). Then the set of solutions to the CE equations of (4) and (5) is exactly the set of solutions to the minimization problem (3).*

The proof is contained in the appendix.

2.2. Consensus equilibrium beyond optimization.

Theorem 1 tells us that CE extends consensus optimization, but as noted above, the novelty of CE is not as a recharacterization of (3) in the case of proximal maps but rather as a framework that applies even when some of the F_i are not proximal mappings and there is *no underlying optimization problem to be solved*. The Plug-and-Play reconstruction method of [29], which yields high quality solutions for important applications in tomography [28] and denoising [25], is, to our knowledge, the first method to use denoisers that do not arise from an optimization for regularized inversion. As we show below, the CE framework also encompasses the Plug-and-Play framework in that if Plug-and-Play converges, then the result is also a CE solution. However, Plug-and-Play grew out of ADMM, and the operators that yield convergence in ADMM are more limited than we would like. Hence, for both consensus optimization and Plug-and-Play priors, CE encompasses the original method but also allows for a wider variety of operators and solution algorithms.

An important point about moving beyond the optimization framework is that a given set of maps F_i may lead to multiple possible CE solutions. This may also happen in the optimization framework when the f_i are not strictly convex since there may be multiple local minima. In the optimization case, the objective function can sometimes be used to select among local minima. The analogous approach for CE is to choose a solution that minimizes the size of $\bar{\mathbf{u}}_\mu^*$, e.g., the L_1 or L_2 norm of $\bar{\mathbf{u}}^*$. This corresponds in some sense to minimizing the tension among the competing forces balanced to find equilibrium.

3. Solving the equilibrium equations.

In this section, we rewrite the CE equations as an unconstrained system of equations and then use this to express the solution in terms of a fixed point problem. We also discuss particular methods of solution, including novel preconditioning methods and methods to solve for a wide range of possible $\mathbf{F}$. We first introduce some additional notation. For $\mathbf{v} \in \mathbb{R}^{nN}$, with $\mathbf{v} = (v_1^T, \dots, v_N^T)$ and each $v_j \in \mathbb{R}^n$, define $\mathbf{F}, \mathbf{G}_\mu : \mathbb{R}^{nN} \to \mathbb{R}^{nN}$ by

$$\mathbf{F}(\mathbf{v}) = \begin{pmatrix} F_1(v_1) \\ \vdots \\ F_N(v_N) \end{pmatrix} \quad \text{and} \quad \mathbf{G}_\mu(\mathbf{v}) = \begin{pmatrix} \bar{\mathbf{v}}_\mu \\ \vdots \\ \bar{\mathbf{v}}_\mu \end{pmatrix}, \tag{7}$$

where $\mathbf{G}_\mu$ has the important interpretation of redistributing the weighted average of the vector components given by $\bar{\mathbf{v}}_\mu = \sum_{i=1}^N \mu_i v_i$ across each of the output components.

Also, for $x \in \mathbb{R}^n$, let $\hat{x}$ denote the vector obtained by stacking N copies of x. With this notation, the CE equations are given by

$$\begin{aligned} \mathbf{F}(\hat{x}^* + \mathbf{u}^*) &= \hat{x}^*, \\ \bar{\mathbf{u}}^*_\mu &= \mathbf{0}. \end{aligned} \tag{8}$$

This notation allows us to reformulate the CE equations as the solution to a system of equations.

Theorem 2. *The point* $(x^*, \mathbf{u}^*)$ *is a solution of the CE equations* (4) *and* (5) *if and only if the point* $\mathbf{v}^* = \hat{x}^* + \mathbf{u}^*$ *satisfies* $\bar{\mathbf{v}}^*_\mu = x^*$ *and*

$$\mathbf{F}(\mathbf{v}^*) = \mathbf{G}_\mu(\mathbf{v}^*). \tag{9}$$

Proof. Let $(x^*, \mathbf{u}^*)$ be a solution to the CE equations, and let $\mathbf{v}^* = \hat{x}^* + \mathbf{u}^*$. Linearity of $\mathbf{G}_\mu$ together with $\bar{\mathbf{u}}^*_\mu = \mathbf{0}$ gives $\mathbf{G}_\mu(\mathbf{v}^*) = \hat{x}^*$, so in particular, $\bar{\mathbf{v}}^*_\mu = x^*$. Using this in (8) gives (9).

Conversely, if $\mathbf{v}^*$ satisfies (9), define $x^* = \bar{\mathbf{v}}^*_\mu$ and $\mathbf{u}^* = \mathbf{v}^* - \hat{x}^*$. Then (4) and (5) are satisfied by definition of x^* and (9). ■

We use this to reformulate CE as a fixed point problem.

Corollary 3 (consensus equilibrium as fixed point). *The point* $(x^*, \mathbf{u}^*)$ *is a solution of the CE equations* (4) *and* (5) *if and only if the point* $\mathbf{v}^* = \hat{x}^* + \mathbf{u}^*$ *satisfies* $\bar{\mathbf{v}}^*_\mu = x^*$ *and*

$$(2\mathbf{G}_\mu - \mathbf{I})(2\mathbf{F} - \mathbf{I})\mathbf{v}^* = \mathbf{v}^*. \tag{10}$$

When $\mathbf{F}$ is a proximal map for a function f, then $2\mathbf{F} - \mathbf{I}$ is known as the reflected resolvent of f. Discussion and results concerning this operator can be found in [2, 12, 15] among many other places. This fixed point formulation is closely related to the fixed point formulation for minimizing the sum of two functions using DR splitting; this is seen clearly in section 4 of [13] among other places. The form given here is somewhat different in that the reflected resolvents are computed in parallel and then averaged, as opposed to the standard sequential form. Beyond that, the novelty here is in the equivalence of this formulation with the CE formulation.

Proof of Corollary 3. By Theorem 2, $(x^*, \mathbf{u}^*)$ is a solution of (4) and (5) if and only if $\mathbf{v}^* = \hat{x}^* + \mathbf{u}^*$ satisfies $\bar{\mathbf{v}}^* = x^*$ and (9). From (9) we have $(2\mathbf{F} - \mathbf{I})\mathbf{v}^* = (2\mathbf{G}_\mu - \mathbf{I})\mathbf{v}^*$. A calculation shows that $\mathbf{G}_\mu \mathbf{G}_\mu = \mathbf{G}_\mu$, so $(2\mathbf{G}_\mu - \mathbf{I})(2\mathbf{G}_\mu - \mathbf{I}) = \mathbf{I}$ by linearity of $\mathbf{G}_\mu$. Hence applying $2\mathbf{G}_\mu - \mathbf{I}$ to both sides gives (10). Reversing these steps returns from (10) to (9). ■

3.1. Anisotropic preconditioned Mann iteration for nonexpansive maps.

Define $\mathbf{T} = (2\mathbf{G}_\mu - \mathbf{I})(2\mathbf{F} - \mathbf{I})$. When $\mathbf{T}$ is nonexpansive and has a fixed point, we can use Mann iteration to find a fixed point of $\mathbf{T}$ as required by (10). For a fixed parameter $\rho \in (0, 1)$, this takes the form

$$\mathbf{w}^{k+1} = (1 - \rho)\mathbf{w}^k + \rho \mathbf{T}(\mathbf{w}^k) \tag{11}$$

with iterates guaranteed to converge to a fixed point of $\mathbf{T}$. In the context of minimization problems in which $\mathbf{F}$ and $\mathbf{G}$ are both proximal maps, and depending on the choice of ρ, iterations of this form are essentially variants of the proximal point algorithm and give rise to the (generalized) DR algorithm, the Peaceman–Rachford algorithm, and the ADMM algorithm, including overrelaxed and underrelaxed variants of ADMM. In the case of $N = 2$ and $\rho = 0.5$, the form in (11) is equivalent up to a change of variables to the standard ADMM algorithm; other values of ρ give overrelaxed and underrelaxed variants. Early work in this direction appears in [11] and [12]. A concise discussion is found in [15], which together with [14] provides a preconditioned version of this algorithm in the case of $N = 2$. This preconditioning is obtained by replacing the original minimization of $f(x)+g(x)$ by minimization of $f(Dq)+g(Dq)$, which gives rise to iterations involving the conjugate proximal maps $D^{-1}F_D(Dq)$, where F_D is the proximal map for f as in (6) using the norm $\|\cdot\|_{(DD^T)^{-1}}$ in place of the usual Euclidean metric. [15] includes some results about the rate of convergence as a function of D. In some cases, a larger value of ρ leads to faster convergence relative to $\rho = 0.5$. There are also results on convergence in the case that fixed ρ is replaced by a sequence of ρ_k such that $\sum_k \rho_k(1-\rho_k) = \infty$ [2]. Further discussion and early work on this approach is found in [11, 12]. With some abuse of nomenclature, we use ADMM below to refer to Mann iteration with $\rho = 0.5$.

Here we describe an alternative preconditioning approach for Mann iteration in which we use an invertible linear map $\mathbf{H}$ in place of the scalar ρ in (11). In this approach, $\mathbf{T}$ can be any nonexpansive map, and $\mathbf{H}$ can be any symmetric matrix with $\mathbf{H}$ and $\mathbf{I}-\mathbf{H}$ both positive definite.

Theorem 4. *Let $\mathbf{H}$ be a positive definite, symmetric matrix, and let $\mathbf{T}$ be nonexpansive on $\mathbb{R}^{nN}$ with at least one fixed point. Suppose that the largest eigenvalue of $\mathbf{H}$ is strictly less than* 1. *For any $\mathbf{v}^0$ in $\mathbb{R}^{nN}$, define*

$$\mathbf{v}^{k+1} = (\mathbf{I}-\mathbf{H})\mathbf{v}^k + \mathbf{H}\mathbf{T}(\mathbf{v}^k) \tag{12}$$

for each $k \geq 0$. Then the sequence $\{\mathbf{v}^k\}$ converges to a fixed point of $\mathbf{T}$.

The idea of the proof is similar to the proof of convergence for Mann iteration given in [26], but using a norm that weights differently the orthogonal components arising from the spectral decomposition of $\mathbf{H}$. The proof is contained in the appendix.

We note that in the case that each f_i is a proper, closed, convex function on $\mathbb{R}^n$ and F_i is the proximal map as in (6), then the map $2\mathbf{F}-\mathbf{I}$ is nonexpansive, so this preconditioning method can be used to find a solution to the problem in (3). The asymptotic rate of convergence with this method is not significantly different from that obtained with the isotropic scaling obtained with a scalar ρ. However, we have found this approach to be useful for accelerating convergence in certain tomography problems in which various frequency components converge at different rates, leading sometimes to visible oscillations in the reconstructions as a function of iteration number. An appropriate choice of the preconditioner H can dampen these oscillations and provide faster convergence in the initial few iterations. We will explore this example and related algorithmic considerations further in a future paper.

3.2. Beyond nonexpansive maps.

The iterative algorithms obtained from (11) and (12) give guaranteed global convergence when T is nonexpansive and ρ (or H) satisfy appropriate conditions. However, the iterates of (11) may still be convergent for more general maps T. We illustrate this behavior in Case 1 of section 4.2.

In fact, when T is differentiable at a fixed point, the rate of convergence is closely related to the spectrum of the linearization of T near this fixed point. The parameter ρ in (11) maintains a fixed point at $\mathbf{w}^* = T(\mathbf{w}^*)$ but changes the linear part of the iterated map to have eigenvalues $\mu_j = \rho\lambda_j + (1-\rho)$, where $\lambda_1, \ldots, \lambda_n$ are the eigenvalues of the linear part of T. The iterates of (11) converge locally exactly when all of these μ_j are strictly inside the unit disk in the complex plane. This can be achieved for sufficiently small ρ precisely when the real part of each λ_j is less than 1. Since there is no constraint on the complex part of the eigenvalues, the map T may be quite expansive in some directions. In this case, the optimal rate of convergence is obtained when ρ is chosen so that the eigenvalues μ_j all lie within a minimum radius disk about the origin.

The use of ρ to affect convergence rate and/or to promote convergence is closely related to the ideas of overrelaxation and underrelaxation as applied in a variety of contexts. See, e.g., [16] for further discussion in the context of linear systems. In the current setting, the use of $\rho < 1/2$ is a form of underrelaxation that is related to methods for iteratively solving ill-posed linear systems. In the following theorem, the main idea is to make use of underrelaxation in order to shrink the eigenvalues of the resulting operator to the unit disk and thus guarantee convergence.

Theorem 5 (local convergence of Mann iterates). *Let $\mathbf{F} : \mathbb{R}^n \to \mathbb{R}^n$ and $\mathbf{G} : \mathbb{R}^n \to \mathbb{R}^n$ be maps such that $\mathbf{T} = (2\mathbf{G} - \mathbf{I})(2\mathbf{F} - \mathbf{I})$ has a fixed point $\mathbf{w}^*$. Suppose that $\mathbf{T}$ is differentiable at $\mathbf{w}^*$ and that the Jacobian of $\mathbf{T}$ at $\mathbf{w}^*$ has eigenvalues $\lambda_1, \ldots, \lambda_n$ with the real part of λ_j strictly less than 1 for all j. Then there are $\rho \in (0,1)$ and an open set U containing $\mathbf{w}^*$ such that for any initial point $\mathbf{w}^0$ in U, the iterates defined by (11) converge to $\mathbf{w}^*$.*

The proof of this theorem is given in Appendix A.

3.3. Newton's method.

By formulating the CE as a solution to $\mathbf{F}(\mathbf{v}) - \mathbf{G}_\mu(\mathbf{v}) = 0$, we can apply a variety of root-finding methods to find solutions. Likewise, rewriting (10) as $T(\mathbf{v}) - \mathbf{v} = 0$ gives the same set of options.

Let H be a smooth map from $\mathbb{R}^n$ to $\mathbb{R}^n$. The basic form of Newton's method for solving $H(x) = 0$ is to start with a vector x_0 and look for a vector dx to solve $H(x_0 + dx) = 0$. A first-order approximation gives $J_H(x_0)dx = -H(x_0)$, where $J_H(x_0)$ is the Jacobian of H at x_0. If this Jacobian is invertible, this equation can be solved for dx to give $x_1 = x_0 + dx$ and the method iterated. There are a wide variety of results concerning the convergence of this method with and without preconditioning, with various inexact steps, etc. An overview and further references are available in [20].

For large scale problems, calculating the Jacobian can be prohibitively expensive. The Jacobian-free Newton–Krylov (JFNK) method is one approach to avoid the need for a full calculation of the Jacobian. Let $J = J_H(x_0)$. The key idea in Newton–Krylov methods is that instead of trying to solve $Jdx = -H(x_0)$ exactly, we instead minimize $\|H(x_0) + Jdx\|$

over the vectors dx in a Krylov subspace, K_j. This subspace is defined by first calculating the residual $r = -H(x_0)$ and then taking

$$K_j = \operatorname{span}\{r, Jr, \dots, J^{j-1}r\}.$$

The basis in this form is typically highly ill conditioned, so the generalized minimal residual method is often used to produce an orthonormal basis and solve the minimization problem over this subspace. This form requires only multiplication of a vector by the Jacobian, which can be approximated as

$$Jr \approx \frac{H(x_0 + \epsilon r) - H(x_0)}{\epsilon}.$$

Applying this to produce K_j requires j applications of the map H together with the creation of the Arnoldi basis elements, which can then be used to find the minimizing dx by reducing to a standard least squares problem of dimension j. Various stopping conditions can be used to determine an appropriate j. These calculations take the place of the solution of $Jdx = -H(x_0)$. In cases for which there are many contracting directions and only a few expanding directions for Newton's method near the solution point, the JFNK method can be quite efficient. A more complete description with a discussion of the benefits and pitfalls, approaches to preconditioning, and many further references is contained in [18].

We note in connection with the previous section that if $\mathbf{H}$ is chosen to be $(\mathbf{I} - J_{\mathbf{T}}(\mathbf{v}^k))^{-1}$ in (12), then the choice of $\mathbf{v}^{k+1}$ in Theorem 4 is an exact Newton step applied to $\mathbf{I} - \mathbf{T}$. That is, the formula for the step in Newton's method in this case is

$$\mathbf{v}^{k+1} - \mathbf{v}^k = -(\mathbf{I} - J_T(v^k))^{-1}(\mathbf{I} - \mathbf{T})\mathbf{v}^k$$

or

$$\mathbf{v}^{k+1} = (\mathbf{I} - \mathbf{H})\mathbf{v}^k + \mathbf{H}\mathbf{T}\mathbf{v}^k,$$

which is the same as the formula in Theorem 4.

In the examples below, we use standard Newton's method applied to both $\mathbf{F} - \mathbf{G}$ and $\mathbf{T} - \mathbf{I}$ in the first example and JFNK applied to $\mathbf{F} - \mathbf{G}$ in the second. Because of the connection with Mann iteration just given, we use the term Newton Mann to describe Newton's method applied to $\mathbf{T} - \mathbf{I}$.

3.4. Other approaches.

An alternative approach is to convert the CE equations back into an optimization framework by considering the residual error norm given by

$$R(\mathbf{v}) \overset{\Delta}{=} \|\mathbf{F}(\mathbf{v}) - \mathbf{G}_\mu(\mathbf{v})\| \tag{13}$$

and minimizing $R^2(\mathbf{v})$ over $\mathbf{v}$. Assuming that a solution of the CE equations exists, then that solution is also a minimum of this objective function. In the case that $\mathbf{F}$ is twice continuously differentiable, a calculation using the facts that $R(\mathbf{v}^*) = 0$ and that $\mathbf{G}_\mu$ is linear shows that the Hessian of $R^2(\mathbf{v})$ is $2A^TA + O(\|\mathbf{v} - \mathbf{v}^*\|)$, where $A\mathbf{v} = J_{\mathbf{F}}(\mathbf{v}^*)\mathbf{v} - \mathbf{G}_\mu\mathbf{v}$. Hence $R^2(\mathbf{v})$ is locally convex near $\mathbf{v}^*$ as long as A has no eigenvalue equal to 0. Since $\mathbf{G}_\mu$ is a projection, its only eigenvalues are 0 and 1; hence this is equivalent to saying that $J_{\mathbf{F}}(\mathbf{v}^*)$ does not have 1 as an eigenvalue. If A does have an eigenvalue 0, then a perturbation of the form $\mathbf{F}_\epsilon(\mathbf{v}) = \mathbf{F}(\mathbf{v}) + \epsilon\mathbf{v}$ produces a unique solution, which can be followed in a homotopy scheme as ϵ decreases to 0.

One possible algorithm for this approach is the Gauss–Newton method, which can be used to minimize a sum of squared function values and which does not require second derivatives.

We note that the residual error of equation (13) is also useful as a general measure of convergence when computing the CE solution; we use this in plots below.

Other candidate solution algorithms include the forward-backward algorithm and related algorithms as presented in [9]. We leave further investigation of efficient algorithms for solving the CE equations to future research.

4. Experimental results.

Here we provide some computational examples of varying complexity. For each of these examples, at least one of the component maps F_i is not a proximal mapping, so the traditional optimization formulation of (1) or (3) is not applicable.

We start with a toy model in two dimensions to illustrate the ideas, then follow with some more complex examples.

4.1. Toy model.

In this example we have $v_1 = (v_{11}, v_{12})^T$, $v_2 = (v_{21}, v_{22})^T$, both in $\mathbb{R}^2$, and maps F_1 and F_2 defined by

$$F_1(v_1) = \left(I + \sigma^2 A^T A\right)^{-1} (v_1 + \sigma^2 A^T y),$$
$$F_2(v_2) = 1.1(v_{21} + 0.2, v_{22} - 0.2\sin(2v_{22}))^T.$$

In this case, F_1 is a proximal map as in (6) corresponding to $f(x) = \|Ax - y\|^2/2$, and F_2 is a weakly expanding map designed to illustrate the properties of CE. We use $\sigma = 1$ and

$$A = \begin{bmatrix} 0.3 & 0.6 \\ 0.4 & 0.5 \end{bmatrix}, \quad y = \begin{bmatrix} 1 \\ 1 \end{bmatrix}.$$

We take $\mu_1 = \mu_2 = 0.5$ and so write G for G_μ. We apply Newton iterations to $\mathbf{F}(\mathbf{v}) - \mathbf{G}(\mathbf{v}) = 0$ and to the fixed point formulation $T(\mathbf{v}) - \mathbf{v} = 0$. In both cases, the Jacobian of F_2 is evaluated only at the initial point.

Figure 1 shows the vectors obtained from each of the maps F_1 and F_2. Blue line segments are vectors from a point v_1 to $F_1(v_1)$, and red line segments are vectors from a point v_2 to $F_2(v_2)$. The starting points of each pair of red and blue vectors are chosen so that they have a common ending point, signified by a black dot. Open squares show the trajectories of v_1^k in blue and v_2^k in red. The trajectories converge to points for which the corresponding red and blue vectors have a common endpoint and are equal in magnitude and opposite in direction; this is the CE solution. The plots shown are for Newton's method applied to $\mathbf{F} - \mathbf{G}$; the plots for Newton's method applied to $\mathbf{T} - \mathbf{I}$ are similar (not shown). In the right panel of this figure, we use the true fixed point to plot error versus iterate for this example using all three methods. The expansion in F_2 prevents ADMM from converging in this example.

4.2. Stochastic matrix.

The next example uses the proximal map form for F_1 as in the previous example, although now with dimension $n = 100$. A and y were chosen using the random number generator rand in MATLAB, approximating the uniform distribution on $[0, 1]$ in each component. The map F_2 has the form $F_2(v) = Wv$; here W is constructed by first choosing entries at random in the interval $[0, 1]$ as for A, then replacing the diagonal entry by the maximum entry in that row (in which case the maximum entry may appear twice in

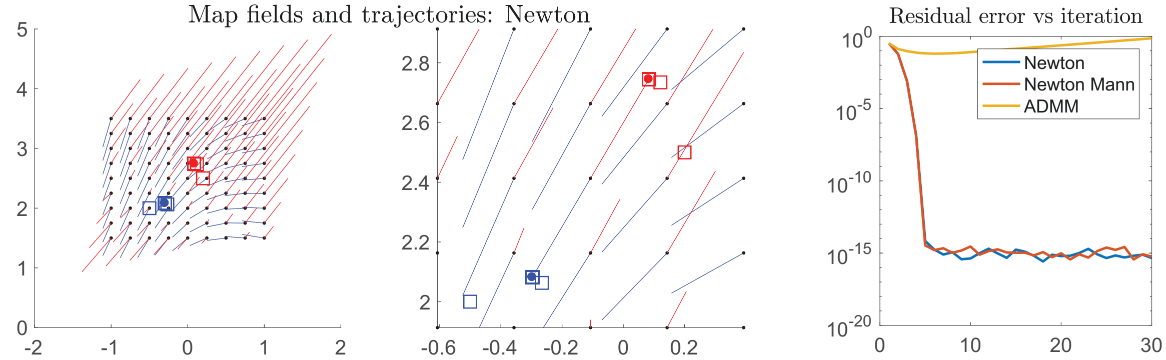


Figure 1. *Left: Map fields and trajectories for a two-dimensional toy example using Newton's method applied to solve* $\mathbf{F}(\mathbf{v}) - \mathbf{G}(\mathbf{v}) = 0$*. Blue segments show the map* $v_1 \mapsto F_1(v_1)$*, red segments show* $v_2 \mapsto F_2(v_2)$*, and black dots show the common endpoints of these maps. Blue and red open squares show the points* v_1^k *and* v_2^k*, respectively. Filled red and blue circles show the CE solution. Middle: Zoom in near the fixed point of the plot on the left. Right: Error* $\|\mathbf{F}(\mathbf{v}^k) - \mathbf{G}(\mathbf{v}^k)\|$ *as a function of iteration for Newton's method applied to* $\mathbf{F} - \mathbf{G}$*, Newton's method applied to* $\mathbf{T} - \mathbf{I}$ *(labeled as Newton Mann), and standard Mann iteration with* $\rho = 0.5$ *(labeled as ADMM).*

one row), and then normalizing so that each row sums to 1. This mimics some of the features in a weight matrix appearing in denoisers such as nonlocal means [6] but is designed to allow us to compute an exact analytic solution of the CE equations. In particular, since W is not symmetric, F_2 cannot be a proximal map, as shown in [28].

In order to illustrate possible convergence behaviors, we first fix the matrices A and W and the vector y as above and then use a one-parameter family of maps $F_{2,r}(v) = rWv + (1-r)I/2$. When $0 \le r \le 1$, this map averages W and $I/2$. The map $I/2$ is a proximal map as in (6) with $\sigma = 1$ and $f_i(v) = \|v\|^2/2$, i.e., the proximal map associated with a quadratic regularization term. In the framework of Corollary 3, the map $F_{2,r}$ satisfies $2F_{2,r}(v) - v = r(2W - I)v$. Hence the scaling of r controls the expansiveness of one of the component maps in $2\mathbf{F} - \mathbf{I}$, and hence the expansiveness of the operator in (10) through the averaging operator G_μ. For the examples here, we choose r to be 1.02 and 1.06. As described below, with appropriate choices of parameters, the JFNK method converges for both examples, while ADMM converges for the first one only.

Recall that if the Lipschitz constant, $L(T)$, is strictly less than 1, then the operator T is a contraction, and if $L(T) \le 1$, we say it is nonexpansive. Moreover, for linear operators, $L(T) = \sigma_{max}$, where σ_{max} is the maximum singular value of T, and $\sigma_{max} \ge |\lambda_{max}|$, where λ_{max} is the eigenvalue with greatest magnitude.

Case 1, $r = 1.02$. In this case, $\mathbf{T}$ has Lipschitz constant $L(\mathbf{T}) > 1$, and the conditions of Theorem 4 or similar theorems on the convergence of Mann iteration for the convergence of nonexpansive maps do not hold. However, in this case, $\mathbf{T}$ is affine (linear map plus constant), and all eigenvalues of the linear part of $(\mathbf{T} + \mathbf{I})/2$ lie strictly inside the unit circle. From (11) with $\rho = 1/2$ and basic linear algebra, this means that Mann iteration converges. This is confirmed in Figure 2. In this example, convergence for Mann iteration can be improved by taking ρ to be 0.8, in which case the convergence

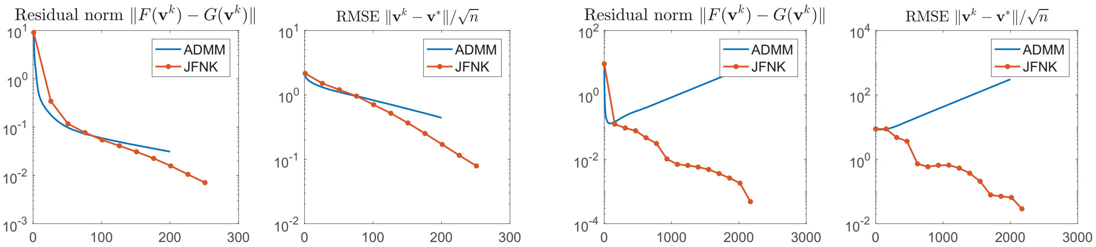


Figure 2. *Residual norm $\|F(\mathbf{v}^k) - \mathbf{G}(\mathbf{v}^k)\|$ and root mean square error $\|v^k - v^*\|/\sqrt{n}$ versus iteration. Left two panels (Case 1): T has Lipschitz constant larger than 1, but all eigenvalues have real part strictly less than 1. Both methods converge. Right two panels (Case 2): T has an eigenvalue with real part larger than 1. JFNK converges, while ADMM diverges.*

is marginally better than that for JFNK. For this example, we used a Krylov subspace of dimension 10, so that each Newton step requires 10 function evaluations. This is indicated by closed circles in the plots.

Case 2, $r = 1.06$. In this case, $\mathbf{T}$ has Lipschitz constant $L(\mathbf{T}) > 1$, and there is an eigenvalue with real part approximately 1.0039, so averaging $\mathbf{T}$ with the identity as in Mann iteration will maintain an eigenvalue larger than 1. In particular, Mann iteration with $\rho = 1/2$ (labelled as ADMM) does not converge, but the JFNK algorithm does. For this example, we used a Krylov subspace of dimension 75, so that each Newton step requires 75 function evaluations. This is indicated by closed circles in the plots.

4.3. Image denoising with multiple neural networks.

The third example we give is an image denoising problem using multiple deep neural networks. This problem is more complex in that we use several neural networks, none of which is tuned to match the noise in the image to be denoised. Nevertheless, we show that CE is often able to outperform each individual network. The images and code for this section are available at [1].

The forward model of image denoising is described by the following linear equation:

$$y = x + \eta,$$

where $x \in \mathbb{R}^n$ is latent unknown image, $\eta \sim \mathcal{N}(0, \sigma_\eta^2 I)$ is independently and identically distributed Gaussian noise, and $y \in \mathbb{R}^n$ is the corrupted observation. Our motivation is to find an estimate $x^* \in \mathbb{R}^n$ by solving the CE equation analogous to the classical MAP approach:

$$x^* = \operatorname*{argmin}_{x \in \mathbb{R}^n} \frac{1}{2\sigma_\eta^2} \|y - x\|^2 - \log p(x), \tag{14}$$

where $p(x)$ is the prior of x. However, instead of a prior function, which would induce a proximal map, we will use a set of CNNs, which will play the role of regularization in the way that a prior term does, but which are almost certainly not themselves proximal maps for any function.

To define the CE operators F_i, we consider a set of K image denoisers. Specifically, we use the denoising convolutional neural network (DnCNN) proposed by Zhang et al. [32]. In the code provided by the authors,[1] there are five DnCNNs trained at five different noise levels: $\sigma_1 = 10/255$, $\sigma_2 = 15/255$, $\sigma_3 = 25/255$, $\sigma_4 = 35/255$, and $\sigma_5 = 50/255$. In other words, the user has to choose the appropriate DnCNN to match the actual noise level σ_η. In the CE framework, we see that F_i is the operator

$$F_i(v_i) = \text{DnCNN}(v_i \text{ with denoising strength } \sigma_i). \tag{15}$$

The $(K+1)^{\text{st}}$ CE operator F_{K+1} is the proximal map of the likelihood function:

$$F_{K+1}(v_{K+1}) = \underset{x\in\mathbb{R}^n}{\text{argmin}} \ \frac{1}{2\sigma_\eta^2}\|y-x\|^2 + \frac{1}{2\sigma^2}\|v_{K+1}-x\|^2, \tag{16}$$

where σ is an internal parameter controlling the strength of the regularization $\|v_{K+1}-x\|^2$. In this example, we set $\sigma = \sigma_\eta$ for simplicity.

To make the algorithm more adaptive to the data, we use weights $\mu_i = \frac{p_i}{\sum_{i=1}^{K+1} p_i}$, where

$$p_i = \exp\left\{-\frac{(\sigma_\eta - \sigma_i)^2}{2h^2}\right\} \quad \text{and} \quad p_{K+1} = \sum_{i=1}^{K} p_i. \tag{17}$$

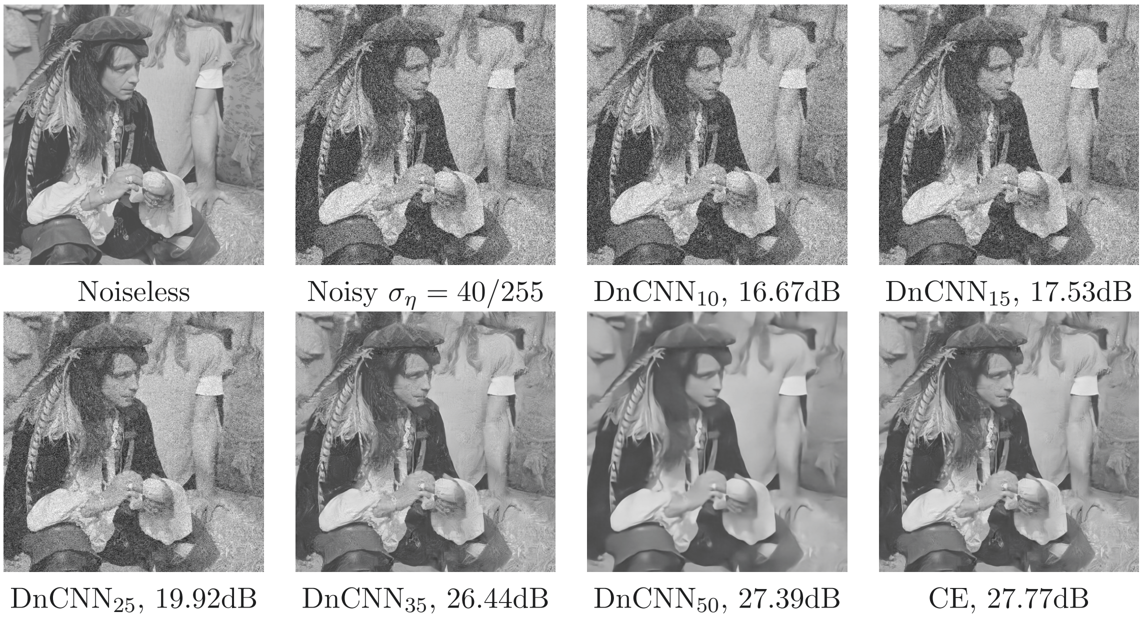
In this pair of equations, p_i measures the deviation between the actual noise level σ_η and the denoising strength of the neural networks σ_i. The parameter $h = 5/255$ controls the cut-off. Therefore, among the five neural networks, p_i weights more heavily the relevant networks. The $(K+1)^{\text{st}}$ weight p_{K+1} is the weight of the map to fit to data. Its value is chosen to be the sum of the weights of the denoisers to provide appropriate balance between the likelihood and the denoisers.

Figures 3 and 4 show some results using noise levels of $\sigma_\eta = 20/255$ and $40/255$, respectively. Notice that none of these noise levels is covered by the trained DnCNNs. Table 1 shows resulting SNR values for the full set of experiments using eight test images and three noise levels of $\sigma_\eta = 20/255, 30/255, 40/255$. The results in the center of the table indicate the result of applying an individual CNN to the noisy image. Because of the form of F_{K+1} in (16) and $\sigma = \sigma_\eta$, the result of this single application of the CNN is the same as the CE solution obtained by using only that single CNN together with F_{K+1}.

Notice that in almost all cases the CE of the full group has the highest peak signal-to-noise ratio (PSNR) when compared to the individual application of the DnCNNs. Also, the improvement in terms of the PSNR is quite substantial for noise levels $\sigma_\eta = 20/255$ and $\sigma_\eta = 30/255$. For $\sigma_\eta = 40/255$, CE still offers PSNR improvement except for House256, which is an image with many smooth regions. In addition, visual inspection of the images shows that the CE result yields the best visual detail while also removing the noise most effectively. While DnCNN denoisers can be very effective, they must be trained in advance using the correct noise level. This demonstrates that the CE can be used to generate a better result by blending together multiple pretrained DnCNNs.

In order to illustrate that the CE solution outperforms a well-chosen linear combination of the outputs from each denoiser, we report a baseline combination result in Table 1. The baseline results are generated by

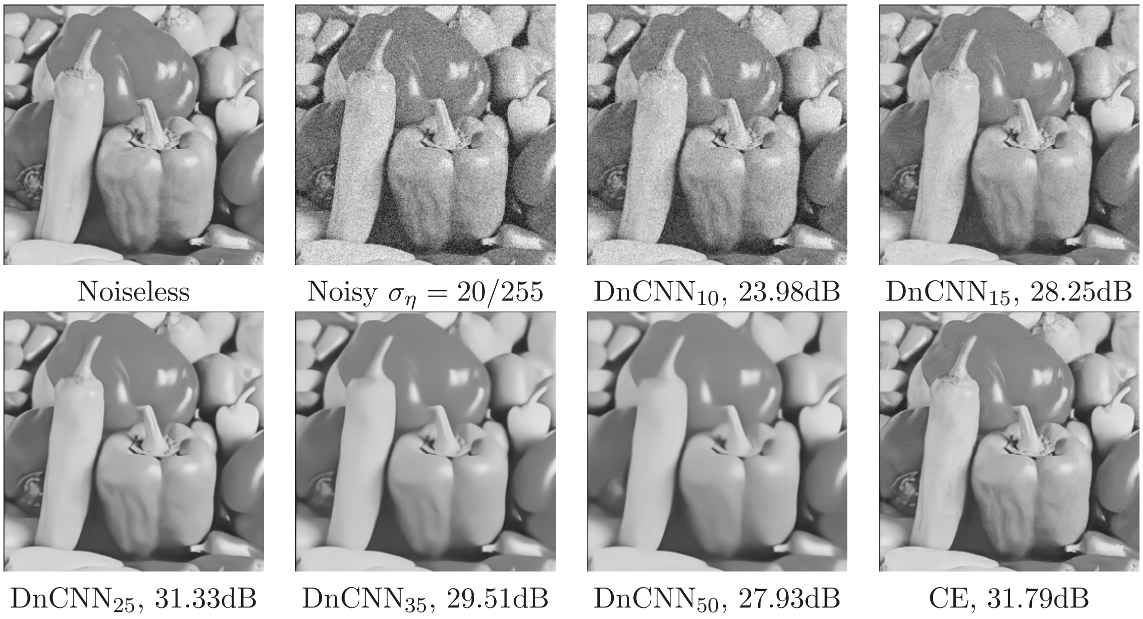
[1] Code available at https://github.com/cszn/ircnn.



Noiseless | Noisy $\sigma_\eta = 40/255$ | DnCNN$_{10}$, 16.67dB | DnCNN$_{15}$, 17.53dB

DnCNN$_{25}$, 19.92dB | DnCNN$_{35}$, 26.44dB | DnCNN$_{50}$, 27.39dB | CE, 27.77dB

Figure 3. *Image denoising experiment for Man512 when $\sigma_\eta = 40/255$. Notice that the CE result has the highest signal-to-noise ratio (SNR) when compared to individual CNN denoisers trained on varying noise levels.*



Noiseless | Noisy $\sigma_\eta = 20/255$ | DnCNN$_{10}$, 23.98dB | DnCNN$_{15}$, 28.25dB

DnCNN$_{25}$, 31.33dB | DnCNN$_{35}$, 29.51dB | DnCNN$_{50}$, 27.93dB | CE, 31.79dB

Figure 4. *Image denoising experiment for Peppers256 when $\sigma_\eta = 20/255$. Notice that the CE result has the highest SNR when compared to individual CNN denoisers trained on varying noise levels.*

Table 1

Image denoising results for actual noise level $\sigma_\eta \in \{20, 30, 40\}/255$.

	DnCNN							Matched
Image	10	15	25	35	50	Baseline	CE	DnCNN
$\sigma = 20/255$								
Barbara512	23.99	28.02	30.49	28.11	25.71	29.80	**30.97**	31.02
Boat512	23.98	27.92	30.61	28.73	27.03	29.86	**31.08**	31.15
Cameraman256	24.12	28.04	30.20	28.52	27.20	29.88	**31.05**	31.07
Hill512	23.93	27.81	30.34	28.68	27.20	29.78	**30.88**	30.92
House256	24.03	28.70	33.70	32.32	30.69	31.38	**33.82**	33.97
Lena512	24.07	28.59	33.06	31.13	29.59	31.12	**33.35**	33.47
Man512	23.94	27.89	30.41	28.46	27.02	29.79	**31.00**	31.08
Peppers256	23.98	28.25	31.33	29.51	27.93	30.27	**31.79**	31.80
$\sigma = 30/255$								
Barbara512	19.49	21.01	26.86	28.49	26.15	28.49	**28.98**	28.86
Boat512	19.48	21.02	26.96	28.92	27.20	28.81	**29.38**	29.36
Cameraman256	19.62	21.14	27.11	28.62	27.23	28.77	**29.18**	29.25
Hill512	19.48	21.01	26.78	28.91	27.38	28.79	**29.35**	29.33
House256	19.44	21.05	28.48	32.17	30.68	31.24	**32.39**	32.32
Lena512	19.49	21.10	28.18	31.37	29.78	30.71	**31.73**	31.69
Man512	19.48	21.01	26.87	28.77	27.21	28.73	**29.28**	29.29
Peppers256	19.48	21.00	27.40	29.45	27.96	29.25	**29.85**	29.81
$\sigma = 40/255$								
Barbara512	16.69	17.54	19.93	26.05	26.51	26.57	**27.14**	27.32
Boat512	16.66	17.52	19.93	26.50	27.36	27.02	**27.82**	28.12
Cameraman256	16.81	17.65	20.13	26.51	27.24	26.98	**27.68**	27.96
Hill512	16.66	17.53	19.92	26.47	27.56	27.05	**27.90**	28.23
House256	16.61	17.50	20.06	28.30	**30.57**	29.00	30.47	31.04
Lena512	16.66	17.55	20.04	28.01	29.88	28.67	**29.95**	30.38
Man512	16.67	17.53	19.92	26.44	27.39	26.99	**27.77**	28.11
Peppers256	16.67	17.53	19.93	26.79	27.87	27.29	**28.09**	28.38

$$\widehat{x}_{\text{baseline}} = \sum_{i=1}^{n} \mu_i \widehat{x}_i,$$

where $\{\widehat{x}_i\}$ are the initial estimates provided by the denoisers and μ_i is defined through 17 without p_{K+1}. That is, we use the same weights as those for CE, excluding the weight for the likelihood term and rescaled to sum to 1 after this exclusion. Therefore, $\widehat{x}_{\text{baseline}}$ can be considered as a linear combination of the initial estimates, with weights defined by the distance between the current noise level and the trained noise levels. The results in Table 1 show that while $\widehat{x}_{\text{baseline}}$ very occasionally outperforms the best of the individual denoisers, it is usually worse than the best individual denoiser and is uniformly worse than CE. In the last column of Table 1 we show the result of DnCNN trained at a noise level matched with the actual noise level. It is interesting to note that CE compares favorably with the matched DnCNN in many cases, except for large sigma, where the matched DnCNN is uniformly better.

We note that [31] uses a linear transformation depending on the noise level of a noisy image in order to match the noise level of a trained neural network, and then applies the inverse linear transformation to the output. This provides another approach to the example above but doesn't include the ability of CE to combine multiple sources of influence without a predetermined conversion from one to the other. We should also point out the recent work of Choi, Elgendy, and Chan [8], which demonstrates an optimal mechanism of combining image denoisers.

5. Conclusion.

We presented a new framework for image reconstruction, which we term CE. The distinguishing feature of the CE solution is that it is defined by a balance among operators rather than the minimum of a cost function. The CE solution is given by the consensus vector that arises simultaneously from the balance of multiple operators, which may include various kinds of image processing operations. In the case of conventional regularized inversion, for which the optimization framework holds, the CE solution agrees with the usual MAP estimate, but CE also applies to a wide array of problems for which there is no corresponding optimization formulation.

We discussed several algorithms for solving the CE equations, including a novel anisotropic preconditioned Mann iteration and a JFNK method. We also introduced a novel precondition method for accelerating the Mann iterations used to solve the CE equations. There is a great deal of room to explore other methods for finding CE solutions as well as for formulating other equilibrium conditions.

Our experimental results, on a variety of problems with varying complexity, demonstrate that the CE approach can solve problems for which there is no corresponding regularized optimization and can in some cases achieve consensus results that are better than any of the individual operators. In particular, we showed how the CE can be used to integrate a number of CNN denoisers, thereby achieving a better result than any individual denoiser.

A. Appendix: Proofs.

Proof of Theorem 1. In order to use $\sigma^2 > 0$ as in (6), we multiply the objective function in (3) by σ^2, which does not change the solution. Define the Lagrangian associated with this scaled problem as

$$L(x, (x_i)_{i=1}^N, (\lambda_i)_{i=1}^N) = \sum_{i=1}^N (\sigma^2 \mu_i f_i(x_i) + (x - x_i)^T \lambda_i),$$

where the $\lambda_i \in \mathbb{R}^n$ are the Lagrange multipliers for the equality constraints $x_i = x$. Since the f_i are all convex and lower-semicontinuous, the first-order KKT conditions are necessary and sufficient for optimality [23, Theorem 28.3]. At a solution point $(x^*, (x_i^*)_{i=1}^N, (\lambda_i^*)_{i=1}^N)$, these conditions are given by

$$\begin{aligned} \nabla_x L(x^*, (x_i^*)_{i=1}^N, (\lambda_i^*)_{i=1}^N) &= 0, \\ \partial_{x_i} L(x^*, (x_i^*)_{i=1}^N, (\lambda_i^*)_{i=1}^N) &\ni 0 \ \forall i = 1, \ldots, N, \\ x_i^* - x^* &= 0 \ \forall i = 1, \ldots, N. \end{aligned}$$

where ∂_{x_i} is the subdifferential with respect to x_i. These convert to

$$\sum_{i=1}^N \lambda_i^* = 0, \tag{18}$$

$$\sigma^2 \mu_i \partial f_i(x_i^*) \ni \lambda_i^* \ \forall i = 1, \ldots, N, \tag{19}$$

$$x_i^* = x^* \ \forall i = 1, \ldots, N. \tag{20}$$

Define $u_i^* = \lambda_i^*/\mu_i$, in which case (18) is the same as (5). Next, use $x_i^* = x^*$ from (20) in (19), and cancel μ_i to get $\sigma^2 \partial f_i(x^*) \ni u_i^*$ for all i. Adding x^* to both sides gives $x^* + \sigma^2 \partial f_i(x^*) \ni x^* + u_i^*$ or

$$(I + \sigma^2 \partial f_i)(x^*) \ni x^* + u_i^*.$$

Since the f_i are convex and $\sigma^2 > 0$, we can invert to get $x^* = (I + \sigma^2 \partial f_i)^{-1}(x^* + u_i^*)$. From [2, Proposition 16.34], this is equivalent to (4) in the case that F_i is the proximal map of (6). ■

Proof of Theorem 4. Since H is symmetric and positive definite, there is an orthogonal matrix Q and a diagonal matrix Λ with $\Lambda_{jj} = \lambda_j > 0$ for all j and $H = Q\Lambda Q^T$. Let q_j be the jth column of Q, and let $\pi_j v = (q_j^T v) q_j$ be orthogonal projection onto the span of q_j. Define the associated norm $\|v\|_j = \|\pi_j v\|$. Also, let λ be the product $\lambda_1 \cdots \lambda_N$, and let $\hat{\lambda}_j = \lambda/\lambda_j$ (i.e., the product of all λ_1 through λ_N except λ_j). Define the weighted norm

$$\|v\|_{H^{-1}}^2 = v^T H^{-1} v = \sum_j \lambda_j^{-1} \|v\|_j^2,$$

which is equivalent to the standard norm on $\mathbb{R}^N$.

By assumption, there is a fixed point $v^* = Tv^*$. Using $\pi_j H = \lambda_j \pi_j$ and applying π_j to both sides of the definition of v^{k+1} gives

$$\begin{aligned}\|v^{k+1} - v^*\|_j^2 = \|&(1-\lambda_j)\pi_j v^k + \lambda_j \pi_j T v^k \\ &- (1-\lambda_j)\pi_j v^* - \lambda_j \pi_j T v^*\|^2.\end{aligned}$$

Here and below, we use $v^* = Tv^*$ freely as needed. As in [26], we use the equality $\|(1-\theta)a + \theta b\|^2 = (1-\theta)\|a\|^2 + \theta\|b\|^2 - \theta(1-\theta)\|a-b\|^2$, which holds for θ between 0 and 1 and can be verified by expanding both sides as a function of θ. In our case, we have $\theta = \lambda_j \in (0,1)$ from the assumptions on H. After conversion back to the norm $\|\cdot\|_j$, this yields

$$\begin{aligned}\|v^{k+1} - v^*\|_j^2 = (1-\lambda_j)\|v^k - v^*\|_j^2 &+ \lambda_j \|Tv^k - Tv^*\|_j^2 \\ &- \lambda_j(1-\lambda_j)\|v^k - Tv^k\|_j^2.\end{aligned}$$

Summing with weights λ_j^{-1} gives

$$\begin{aligned}\sum_j \lambda_j^{-1}\|v^{k+1} - v^*\|_j^2 = \sum_j (\lambda_j^{-1} - 1)\|v^k - v^*\|_j^2 &+ \sum_j \|Tv^k - Tv^*\|_j^2 \\ &- \sum_j \lambda_j^{-1}\lambda_j(1-\lambda_j)\|v^k - Tv^k\|_j^2.\end{aligned}$$

Since T is nonexpansive, the right-hand side is bounded above by replacing $Tv^k - Tv^*$ with $v^k - v^*$ in the second sum. This new sum then exactly cancels the term arising from -1 in the first sum. Let c be the minimum over j of $\lambda_j(1-\lambda_j)$, and note that $c > 0$ since $\lambda_j < 1$ for each j by assumption. Putting these together and re-expressing in the H^{-1} norm gives

$$\|v^{k+1} - v^*\|_{H^{-1}}^2 \le \|v^k - v^*\|_{H^{-1}}^2 - c\|v^k - Tv^k\|_{H^{-1}}^2.$$

The remainder of the proof is nearly identical to that in [26]; we include it for completeness. Iterating in the first term on the right-hand side, we obtain

$$\|v^{k+1} - v^*\|^2_{H^{-1}} \leq \|v^1 - v^*\|^2_{H^{-1}} - c\sum_{i=1}^{k} \|v^j - Tv^j\|^2_{H^{-1}}, \tag{21}$$

and hence

$$\sum_{i=1}^{k} \|v^j - Tv^j\|^2_{H^{-1}} \leq \frac{1}{c}\|v^1 - v^*\|^2_{H^{-1}}.$$

In particular, $\|v^j - Tv^j\|_{H^{-1}}$, and hence $\|v^j - Tv^j\|$ tend to 0 as j tends to ∞. This also implies that

$$\min_{j=1,\ldots,k} \|v^j - Tv^j\|^2_{H^{-1}} \leq \frac{1}{ck}\|v^1 - v^*\|^2_{H^{-1}}.$$

Finally, note that (21) implies that the sequence $\{v^k\}$ is bounded, hence has a limit point, say $\hat{v}$. Since $I - T$ is continuous and $v^k - Tv^k$ converges to 0, we have $\hat{v} = T\hat{v}$. Using $\hat{v}$ in place of v^* in (21), we see that $\|v^k - \hat{v}\|_{H^{-1}}$ decreases monotonically to 0, hence v^k converges to $\hat{v}$. ■

Proof of Theorem 5. Let $\mathbf{T}_\rho$ denote the map $(1-\rho)\mathbf{I} + \rho\mathbf{T}$. Let $\mu_j(\rho) = (1-\rho) + \rho\lambda_j$, and note that $\mathbf{T}_\rho$ has eigenvalues $\mu_1, \ldots, \mu_n$. Since the real part of λ_j is less than 1, the line segment defined by $\mu_j(\rho)$ for ρ in the interval $[0,1]$ has a nonempty intersection with the open unit disk in the complex plane. For each j, there is some $\epsilon_j > 0$ so that this intersection contains the set of points $\mu_j(\rho)$ for ρ in $(0, \epsilon_j]$. Taking ϵ_0 to be the minimum of the ϵ_j and taking ρ in the interval $(0, \epsilon_0]$, there exists $r < 1$ for which $|\mu_j(\rho)| \leq r < 1$ for all j.

For this choice of ρ, let A be the Jacobian of $\mathbf{T}_\rho$ at the fixed point, $\mathbf{w}^*$, which we may assume is the origin. The Schur triangulation gives a unitary matrix Q and an upper triangular matrix U with $U = Q^{-1}AQ$. Write $U = \Lambda + U'$ with Λ diagonal and U' zero on the diagonal. Let $u_{\max}$ be the maximum of $|U'_{i,j}|$ over all entries in U'. For $\epsilon > 0$, define D to be the diagonal $n \times n$ matrix with $D_{i,i} = \epsilon^i$. A computation shows that $D^{-1}UD$ has the same diagonal entries as U but that each off-diagonal has the form $U_{i,j}\epsilon^{j-i}$ with $j > i$, hence is bounded by $\epsilon u_{\max}$ in norm. This plus the differentiability of $\mathbf{T}_\rho$ implies that for x in a neighborhood of 0,

$$\begin{aligned}\|D^{-1}Q^{-1}\mathbf{T}_\rho QDx\| &= \|\Lambda x + D^{-1}U'Dx\| + o(\|x\|)\\ &\leq (r + n\epsilon u_{\max} + R(\|x\|))\|x\|,\end{aligned}$$

where $R(\|x\|)$ decreases to 0 as $\|x\|$ tends to 0. Choosing ϵ and $\|x\|$ sufficiently small, we have $r + n\epsilon u_{\max} + R(\|x\|) < \beta$ for some $\beta < 1$. In this case we can iterate to obtain

$$\|D^{-1}Q^{-1}\mathbf{T}^k_\rho QDx\| \leq \beta^k\|x\|.$$

In other words, for x^0 in a neighborhood N of the origin, the iterates $x^k = D^{-1}Q^{-1}\mathbf{T}^k_\rho QDx^0$ converge geometrically to the origin. Multiplying by QD and labeling $\mathbf{w}^k = QDx^k$, we have $\mathbf{w}^k = \mathbf{T}^k_\rho\mathbf{w}^0$ converges geometrically to 0 for all $\mathbf{w}^0$ in the neighborhood QDN of the origin. ■

Acknowledgments. We thank the referees for many helpful remarks that have improved this paper substantially.

REFERENCES

[1] *MATLAB Implementation for Consensus Equilibrium*, https://engineering.purdue.edu/ChanGroup/code.html. Accessed April 6, 2018.

[2] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, CMS Books Math., Springer, New York, 2011.

[3] D. P. Bertsekas, *Constrained Optimization and Lagrange Multiplier Methods*, Optimization and Neural Computation Series, Athena Scientific, Nashua, NH, 1996.

[4] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Found. Trends Mach. Learn., 3 (2011), pp. 1–122.

[5] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2009.

[6] A. Buades, B. Coll, and J.-M. Morel, *A review of image denoising algorithms, with a new one*, Multiscale Model. Simul., 4 (2005), pp. 490–530.

[7] S. H. Chan, X. Wang, and O. A. Elgendy, *Plug-and-Play ADMM for image restoration: Fixed-point convergence and applications*, IEEE Trans. Comput. Imaging, 3 (2017), pp. 84–98, https://doi.org/10.1109/TCI.2016.2629286.

[8] J. H. Choi, O. A. Elgendy, and S. H. Chan, *Optimal Combination of Image Denoisers*, https://arxiv.org/abs/1711.06712, ArXiv e-print, 2018.

[9] P. L. Combettes and J.-C. Pesquet, *Proximal splitting methods in signal processing*, in Fixed-Point Algorithms for Inverse Problems in Science and Engineering, Springer, New York, 2011, pp. 185–212.

[10] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, *Image denoising by sparse* 3*-D transform-domain collaborative filtering*, IEEE Trans. on Image Process., 16 (2007), pp. 2080–2095, https://doi.org/10.1109/TIP.2007.901238.

[11] J. Eckstein, *Splitting Methods for Monotone Operators with Applications to Parallel Optimization*, Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA, 1989.

[12] J. Eckstein and D. P. Bertsekas, *On the Douglas–Rachford splitting method and the proximal point algorithm for maximal monotone operators*, Math. Program., 55 (1992), pp. 293–318.

[13] P. Giselsson, *Tight global linear convergence rate bounds for Douglas–Rachford splitting*, J. Fixed Point Theory Appl., 19 (2017), pp. 2241–2270, https://doi.org/10.1007/s11784-017-0417-1.

[14] P. Giselsson and S. Boyd, *Preconditioning in fast dual gradient methods*, in Proceedings of the 53rd IEEE Conference on Decision and Control, 2014, pp. 5040–5045, https://doi.org/10.1109/CDC.2014.7040176.

[15] P. Giselsson and S. Boyd, *Metric selection in fast dual forward-backward splitting*, Automatica J. IFAC, 62 (2015), pp. 1–10, https://doi.org/10.1016/j.automatica.2015.09.010.

[16] W. Hackbusch, *Iterative Solution of Large Sparse Systems of Equations*, Appl. Math. Sci. 95, Springer-Verlag, New York, 1994.

[17] U. S. Kamilov, H. Mansour, and B. Wohlberg, *A Plug-and-Play Priors approach for solving nonlinear imaging inverse problems*, IEEE Signal Process. Lett., 24 (2017), pp. 1872–1876, https://doi.org/10.1109/LSP.2017.2763583.

[18] D. Knoll and D. Keyes, *Jacobian-free Newton–Krylov methods: A survey of approaches and applications*, J. Comput. Phys., 193 (2004), pp. 357–397, https://doi.org/10.1016/j.jcp.2003.08.010.

[19] P. L. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979, https://doi.org/10.1137/0716071.

[20] J. Nocedal and S. J. Wright, *Numerical Optimization*, 2nd ed., Springer, New York, 2006.

[21] S. Ono, *Primal-dual Plug-and-Play image restoration*, IEEE Signal Process. Lett., 24 (2017), pp. 1108–1112, https://doi.org/10.1109/LSP.2017.2710233.

[22] E. T. Reehorst and P. Schniter, *Regularization by Denoising: Clarifications and New Interpretations*, preprint, arXiv:1806.02296, 2018.

[23] R. Rockafellar, *Convex Analysis*, Princeton University Press, Princeton, N.J., 1997, c1970.

[24] Y. Romano, M. Elad, and P. Milanfar, *The little engine that could: Regularization by denoising (RED)*, SIAM J. Imaging Sci., 10 (2017), pp. 1804–1844.

[25] A. Rond, R. Giryes, and M. Elad, *Poisson inverse problems by the Plug-and-Play scheme*, J. Vis. Commun. Image Represent., 41 (2016), pp. 96–108, https://doi.org/10.1016/j.jvcir.2016.09.009.

[26] E. Ryu and S. Boyd, *A primer on monotone operator methods survey*, Appl. Comput. Math., 15 (2016), pp. 3–43.

[27] S. Sreehari, S. V. Venkatakrishnan, K. L. Bouman, J. P. Simmons, L. F. Drummy, and C. A. Bouman, *Multi-resolution data fusion for super-resolution electron microscopy*, in Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), July 2017, pp. 1084–1092, https://doi.org/10.1109/CVPRW.2017.146.

[28] S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, G. T. Buzzard, L. F. Drummy, J. P. Simmons, and C. A. Bouman, *Plug-and-Play priors for bright field electron tomography and sparse interpolation*, IEEE Trans. Comput. Imaging, 2 (2016), pp. 408–423, https://doi.org/10.1109/TCI.2016.2599778.

[29] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, *Plug-and-Play priors for model based reconstruction*, in Proceedings of the IEEE Global Conference on Signal and Information Processing (GlobalSIP), 2013, IEEE, pp. 945–948.

[30] S. V. Venkatakrishnan, L. F. Drummy, M. De Graef, J. P. Simmons, and C. A. Bouman, *Model based iterative reconstruction for bright field electron tomography*, in Proceedings of the IS&T/SPIE Symposium on Electronic Imaging, 2013, p. 86570A.

[31] Y. Q. Wang and J. M. Morel, *Can a single image denoising neural network handle all levels of Gaussian noise?*, IEEE Signal Process. Lett., 21 (2014), pp. 1150–1153, https://doi.org/10.1109/LSP.2014.2314613.

[32] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, *Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising*, IEEE Trans. Image Process., 26 (2017), pp. 3142–3155, https://doi.org/10.1109/TIP.2017.2662206.